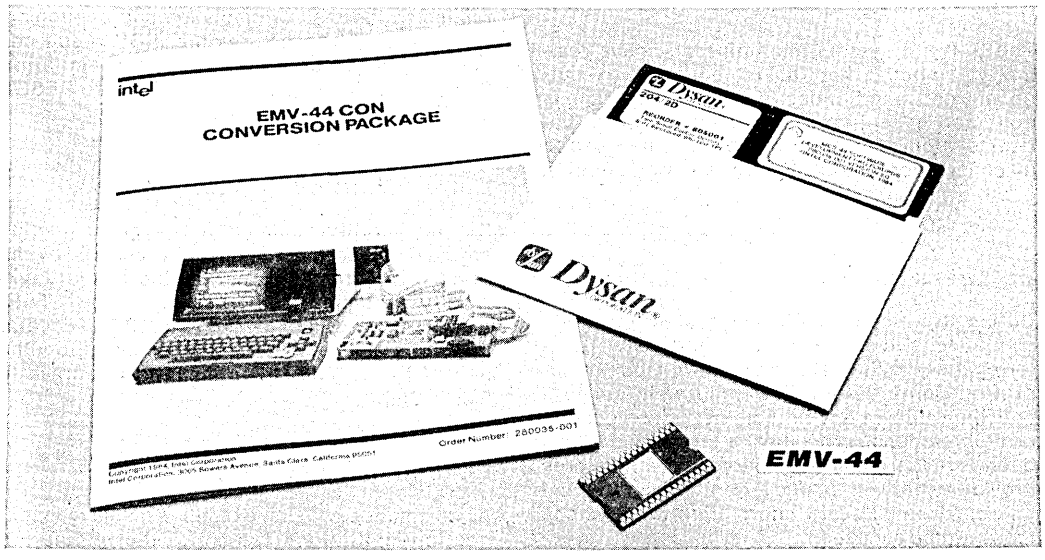
EMV-44 CON
8044 EMULATION VEHICLE
CONVERSION PACKAGE

- **All materials needed to add 8044 support to an EMV-51/51A system**
- **Full-speed, real-time 8044 emulation**
  - **Load, drive, timing characteristics**
  - **Full-speed program RAM**
  - **Serial and parallel ports**
  - **SDLC communications port**
- **Breakpoints/trace**
  - **Four execution address breakpoints**
  - **Range, branch, and value breakpoints**
- **Full symbolic debugging, including support for 8044 expanded symbols**
- **Software debugging with or without user system**
- **Advanced ease-of-use features**
  - **Programmable function keys**
  - **Macros**
- **Help facility tailored for 8044 emulation**
- **Hosted on the Intel Personal Development System (iPDS™)**
- **Use to troubleshoot individual 8044-based designs and complete BITBUS™ system**

The EMV-44 conversion package converts an EMV-51 or EMV-51A emulation vehicle to an EMV-44 emulation vehicle. The resulting EMV-44 system interfaces to any user-designed 8044 system and assists in the debugging and development of the system. (The EMV-44 system cannot be purchased as a separate item; to obtain an EMV-44 system, this EMV-44 conversion package must be used to convert either an EMV-51 or EMV-51A system.) The EMV-44 conversion package consists of a special 8044 component, new development software, and new documentation. To create an EMV-44 system, one needs only replace the special 8051 component in the EMV-51 or EMV-51A system with the new 8044 component, and then install the new software. The EMV-44 accurately emulates the electrical and timing characteristics of the user's 8044. A friendly human interface presents commands in a menu display and organizes commands in an easy-to-learn fashion. The EMV-44 system allows the designer to emulate the specified system's 8044 in real-time or single-step mode. Breakpoints allow the user to stop emulation at user-specified conditions, and trace qualifiers allow for conditional display of trace information. Program memory can be displayed and altered using ASM51 mnemonics and symbolic references. Advanced capabilities allow for programmable keys, macros, and control constructs.



JUNE 1984
Order Number:280035-001

FUNCTIONAL DESCRIPTION

The EMV-44 system provides fundamental capabilities for debugging an 8044. These capabilities are described in the following sections.

Real-Time Breakpoint

The EMV-44 system allows the user system to execute user code at full clock speed, until a predefined condition occurs. The breakpoints may be a combination of four execution addresses or a combination of an execution address range and a single execution address. These break capabilities allow the user to stop the user system at various states in the normal processing cycle and to interrogate the state of the system.

Real-Time Memory

The EMV-44 system uses either the EMV-51 system's 4K or the EMV-51A system's 8K of high speed RAM memory.The RAM can be used to execute the user program and allows easy changes to the user code. The RAM memory can be used either in place of the user's memory before the memory exists in the user system or used in lieu of the user's memory to ease the debugging effort.

Real-Time Trace

The EMV-44 system maintains an active real-time trace buffer that tracks the last two executed addresses in the user's system. The trace is collected in real-time during execution of the user system. This information can be used to discover where the user's program was before it broke.

Software Break

During step mode, the EMV-44 system iteratively single steps, then executes a short software interrogation routine. This slow-down mode of operation continues until a register is set to a specific value, or any branch instruction occurs, or until a specified number of instructions have been executed. These additional break features provide users with added execution control and microprocessor state information in exchange for real-time emulation.

Software Trace

The EMV-44 system will, during interrogation, automatically query the 8044 processor and optionally display up to 4 lines of information. This display can show the execution address, disassembled code, current register values, or processor status information.

COMMANDS

The EMV-44 system has a friendly and easy-to-use human interface, and commands that are well organized and easy-to-learn. Menu displays prompt the user and assist in learning the different commands. Sample EMV-44 displays are shown in Figure 1. Commands fall into four categories: utility commands, display/modify commands, emulation commands, and advanced commands. Once these basic command categories are understood, locating any command becomes simple. Table 1 gives a summary of EMV-44 commands and command categories.

The EMV-44 system is a full symbolic emulator, and hence all commands and displays allow for symbolic entry. Thus the EMV-44 system and users communicate by referring explicitly to symbols defined in the user's source program or symbols defined during the debugging session.

Utility Commands

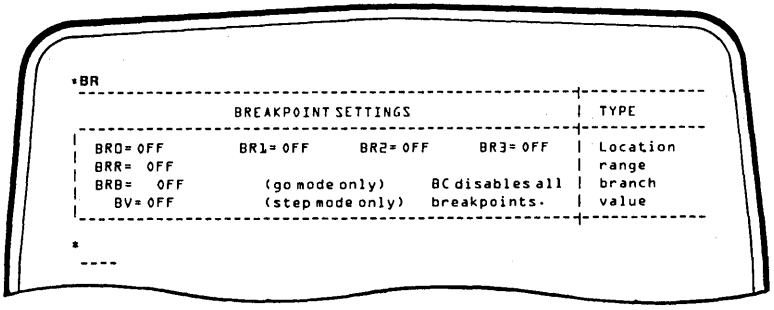
Utility commands perform functions not directly related to the task of emulation and debugging. These commands access the iPDS resources and display information about the emulator. Some examples of utility commands are RESET, LOAD, HELP, and EVALUATE.

Display/Modify Commands

Display/modify commands change or display any register, port, or memory addressable by the 8044 processor chip. Examples of display/-modify commands include REGISTER, ASM/DASM, CBYTE, DBYTE, RBYTE, and PBYTE. A sample of a display resulting from the use of the REGISTER command is shown in Figure 2(a).

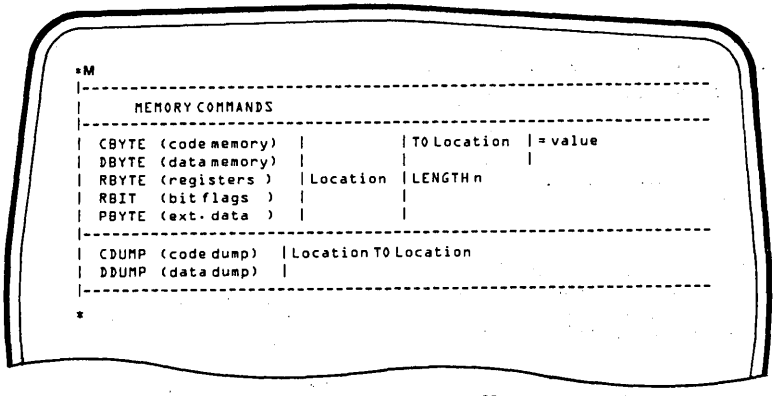
Emulation Commands

All commands causing execution displays, or execution initiation, fall into the emulation



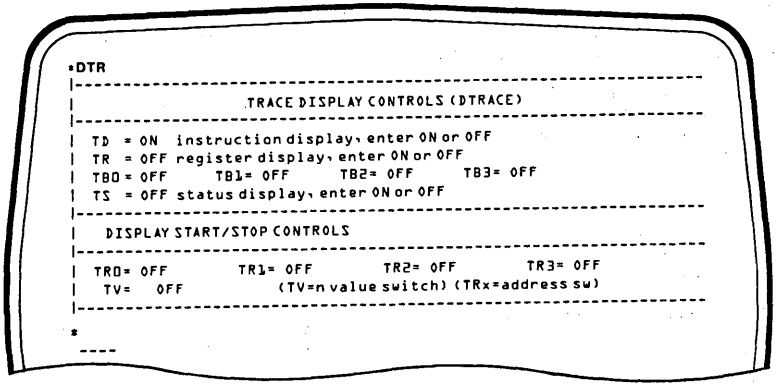
a) Menu Display for Setting Breakpoint

0344



b) Menu Display For Accessing Memory

0330



c) Menu Display For Setting Trace

0348

Figure 1. Typical EMV-44 Menu Displays

280035-001

category. Thus, the GO, BREAK, and TRACE commands are in this category along with BR0,1,2,3, BV, TR0,1,2,3, TS, and STEP.

Advanced Commands

The advanced commands offer the user an easy way to increase the power of the EMV-44 and thus increase the debugging capability of this product. These advanced features allow EMV-44 command sequences to be combined, executed, and stored. Examples of advanced commands include MACRO, FUNCTION, and control constructs. Figure 2(b) shows a display of a macro.

EMULATION MODES

The EMV-44 system combines two approaches to emulation, real-time emulation and software emulation. Programs with time-critical sections of code or critical interrupt routines can be emulated, traced, and debugged in real time.

Real-time emulation supports specific execution breakpoints or range breakpoints. The real-time trace will display up to two instruction addresses last executed. Real-time emulation mode is entered by initiating emulation with the GO command. All break and trace commands associated with the GO command act in real-time emulation mode.

When full-speed emulation is not critical to the debugging effort, the EMV-44 system will emulate one instruction, check for a variety of breakpoint and trace point conditions, display trace information, and continue with another instruction. This slow-down mode of operation provides enhanced break and trace facilities at the expense of non-real-time execution. Slow-down-mode emulation is entered by initiating emulation with the STEP command. Figure 2(a) shows a display for the single-stepping mode.

Table 1. Summary of EMV-44 Commands and Command Categories

Emulation Commands

BREAK - Display breakpoint menu
BR0, 1, 2, 3 - Breakpoint register for execution address
BRR - Breakpoint register for execution range
BRB - Break on branch
BV - Break on value
BC - Clear all breaks
DTRACE - Display trace menu
TB0, 1, 2, 3 - Enable/disable display by bit value
TR0, 1, 2, 3 - Enable/disable display by execution address
TV - Enable/disable display by register value
TR - Enable/disable display of registers
TS - Enable/disable display of PSW
TD - Enable/disable display of code disassembly
STEP - Enter slow-down emulation mode
GO - Enter real-time emulation mode

Advanced Commands

MACRO - Define, and display macro
IF THEN, COUNT, REPEAT, WHILE, UNTIL } Control constructs
FUNCTION - Invoke macro assigned to function key

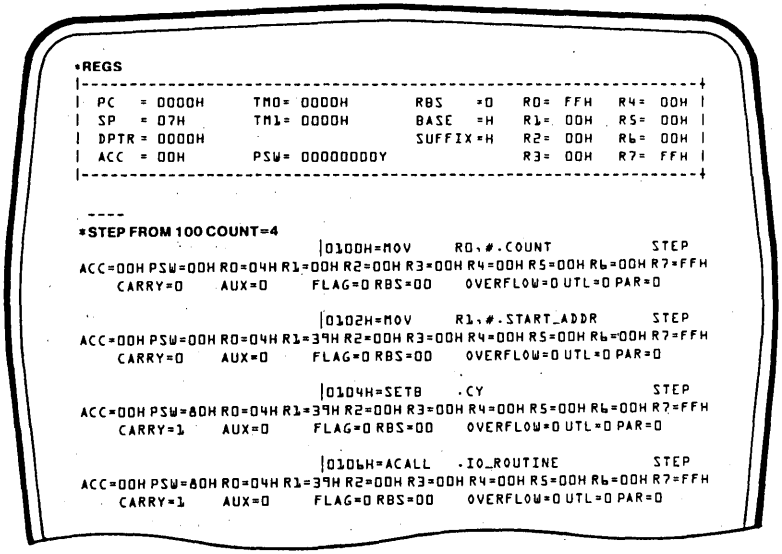
Utility Commands

HELP - Display command syntax
LOAD - Load object file in mapped memory
LIST - Generate copy of emulation work session
DEFINE - Define symbol or macro
SYMBOL - Display symbols
ENABLE/DISABLE - Control for expanded display
EVALUATE - Evaluate any expression
SUFFIX/BASE - Set input and display numeric base
SAVE - Save code memory to file
RESET - Reset emulation processor
EXIT - Terminate EMV-44 session

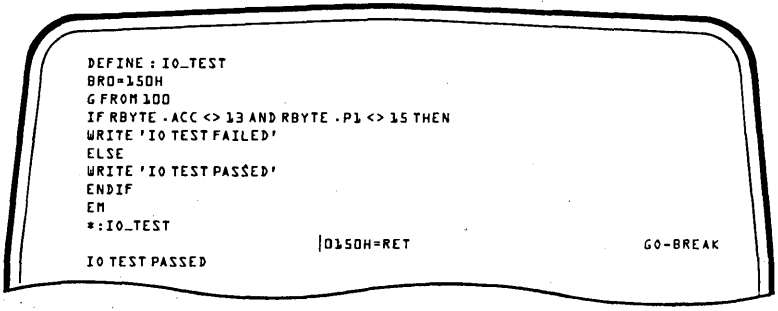
Display/Modify Commands

REGISTER - Change/display 8044 registers
INTERRUPT - Change/display interrupt status
MEMORY - Display menu
CBYTE, DBYTE, PBYTE, RBYTE } Change/display memory
RBIT - Change/display bit memory
CDUMP, DDUMP } Display memory as ASCII and hexadecimal
ASM/DASM - Change/display code memory as assembly language mnemonics

280035-001



a) Display of (1) Registers and (2) Single Stepping through a Portion of a User's Program (Using Symbolics with Selective Trace of Processor and Register Status Information)



b) Display Showing Macro Capability for Debugging System Hardware and Software

1949

Figure 2. Sample Emulation Displays

INTENDED USE

The EMV-44 system is particularly well suited for debugging 8044 designs that include small- to medium-size programs with program complexity that is low to moderate in terms of interrupts, program nesting, and execution flow. In addition to product development, the EMV-44 system is well suited to product testing and servicing. Designs using the BITBUS can be debugged, tested, and serviced while connected to the BITBUS.

FUNCTIONAL DESCRIPTION

The EMV-44 conversion package consists of a special 8044 component, new development software, and new documentation. To create an EMV-44 system, one needs only replace the special 8051 component in the EMV-51 or EMV-51A system with the new 8044 component, and then install the new software. The resulting EMV-44 system has three parts: the controller, the emulator module, and the cable assembly. The controller contains all the logic to support break,

280035-001

trace, emulation, and communication with the host and the emulator module. The emulator module contains the hardware used to execute 8044 code and supplies all MCS®-44 signals to the user's system. This module connects to the controller via a six foot cable, and the controller connects to an iPDS host through the EMV/PROM programming adapter board. This iPDS board is required to use the EMV-44 with the iPDS system.

EMV-44 software contains all the control for user interaction. The software programs the controller, implements all emulator functions, and displays information to the user. This software is run on the iPDS host, and is packaged on a 5-¼ inch diskette. An additional software diagnostic routine, included on the disk, thoroughly checks the EMV-44 hardware.

EMV-44 software will accept and interpret commands entered by the user. These commands will be communicated as a set of micro-commands via a host interface to the controller. Command registers in the controller direct micro-operations to various sections of the break, map, or trace circuitry. Some commands control the emulator board, others determine whether the emulator will emulate the user system, while others interrogate the user system. When appropriate, the controller will pass information back to the host where the information will be processed and displayed to the user. See Fig. 3 for a block diagram of the EMV-44 hardware.

The original EMV-51 or EMV-51A system includes the 8051 Relocating Macro Assembler (ASM51) and the 8051 Linker and Relocater (RL51). The assembler provides full macro capabilities, supports symbolic development for both code development and debugging, and supports modular code development with relocation features. The RL51 utility will relocate, link, and generate loadable object files from the relocatable modules produced by the assembler. EMV-44 fully supports all mnemonics, object file formats, and symbolic references generated by the ASM51 and RL51 programs.

EMV-44 documentation includes a comprehensive user's manual and a command dictionary reference guide.

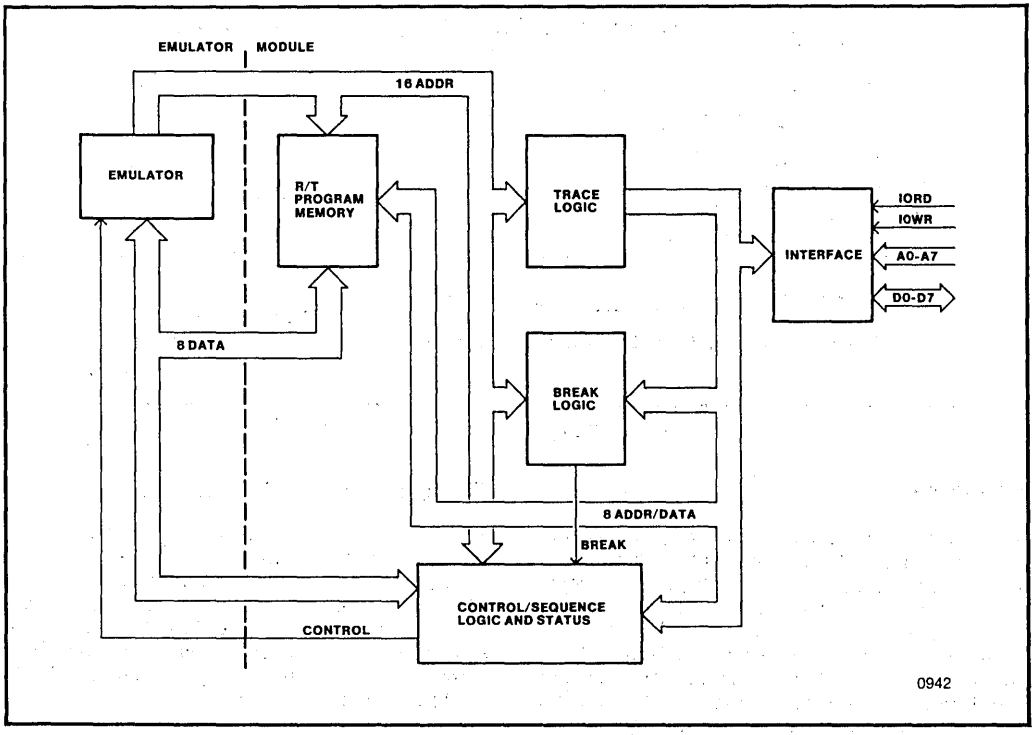


Figure 3. EMV-44 Block Diagram

280035-001

SPECIFICATIONS

EMV-44 Operating Requirements

The EMV-44 system operates with an iPDS system (see Figure 4). The iPDS system must be configured with the EMV/iUP adapter option, iPDS-140.

Equipment Supplied

- 8044 "bondout" microcontroller
- EMV-44 conversion package manual
- EMV-44 software and diagnostic diskette
- EMV-44 label

EMV-44 Emulation Clock Rate

User's system: 1.2 to 12 MHz*
EMV-supplied crystal: 12 MHz

*Note that the bondout 8044 microcontroller supplied with the EMV-44 conversion package has a limitation when serial clock mode 0 is used: the external SCLK signal must be synchronized with the XTAL clock. A simple two flip-flop external circuit can be constructed to provide this synchronization.

EMV-44 Environmental Characteristics

Operating temperature: 0-40° C
Operating humidity: 50-90% RH, non-condensing

EMV-44 Physical Characteristics

Controller: 7.8in. x 1.5 in. x 5.8 in. (19.8 cm. x 3.8 cm. x 14.7 cm.)
Emulator: 3.3 in. x 3.3 in. x 1.5 in. (18.4 cm. x 18.4 cm. x 3.8 cm.)
Total Weight: 1 lb. 7 oz. (0.65 kg.)

EMV-44 Electrical Characteristics

Power requirements from iPDS: +5 V ± 5% @ 1.9A
*Power requirements from user system: +5 V ± 5% @ 200 ma MAX
Characteristics of user socket: Same as 8044, 8344, or 8744

*The emulator can be strapped to draw its power from either the iPDS or the user system.

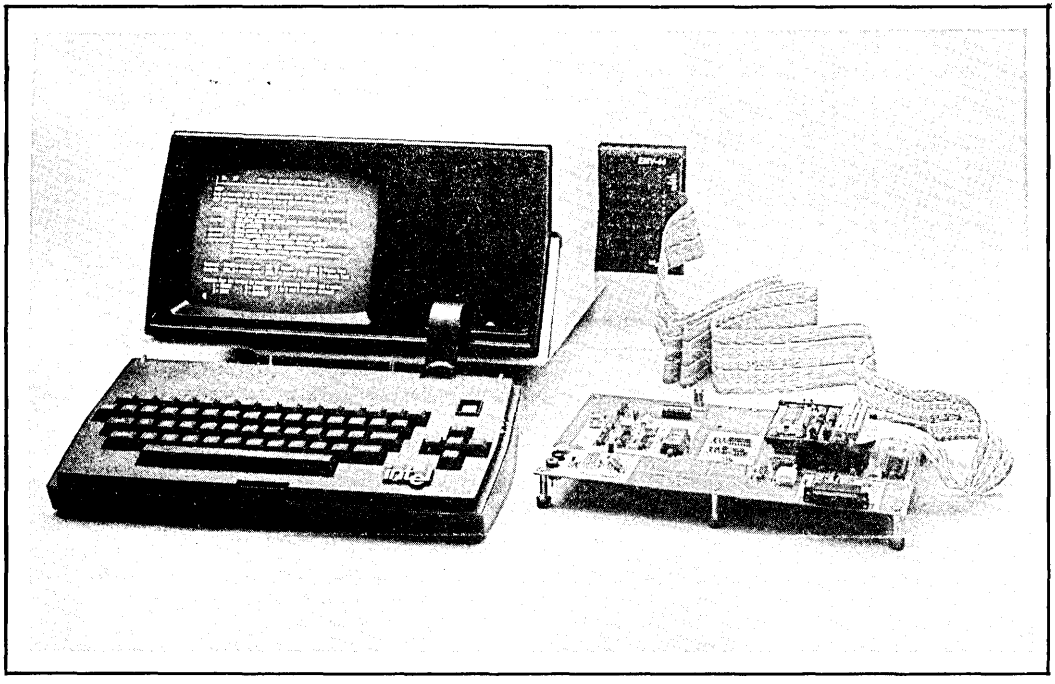


Figure 4. EMV-44 in iPDS™ Debugging Environment

EMV-44 CONVERSION PACKAGE ORDERING INFORMATION

Part Number	Description
iPDS-EMV-44 CON	EMV-44 conversion package with 8044 "bondout" micro-controller, diskette, and documentation